Remarks

In the Office Action, the Examiner noted that claims 1-46 are pending in the application, and that claims 1-46 are rejected. By this amendment, claims 17, 34, and 38 have been amended and new claims 47-52 have been added. Thus, claims 1-52 are pending in the application.

Applicant hereby requests further examination and reconsideration of the application, in view of the foregoing amendments.

In the Specification

In the specification, the table on page 1 has been amended to identify the co-pending applications by their serial numbers. Additionally, three paragraphs were amended to include the serial numbers of three other co-pending applications that were incorporated by reference.

In the Drawings

The Examiner objected to the reference character 100 in Figures 1 and 2 and to 108 in Figure 3. Applicant has changed the labels to address the Examiner's objections.

In the Claims

Rejection Under 35 USC 112 second paragraph

The Examiner rejected claims 38-46 under 35 U.S.C. § 112, second paragraph, as being indefinite, and in particular for insufficient antecedent basis in claim 38. Applicant has amended claim 38 to provide sufficient antecedent basis.

Rejection Under 35 USC 102(b)

The Examiner rejected claims 1-15, 18-19, 21-24, 26-32, and 38-46 under 35 U.S.C. § 102(b), as being anticipated by *Emma, et al.*, U.S. Patent No. 5,353,421 (hereinafter *Emma*). Applicant respectfully traverses the rejection of claims 1-15, 18-19, 21-24, 26-32, and 38-46.

With respect to claim 1, the Examiner asserted that *Emma* teaches selection logic for causing a portion of instruction bytes of an instruction cache line selected by a fetch address not to be provided to an instruction buffer based on offset information specifying a location of a branch instruction within the cache line. For the following reasons, Applicant respectfully asserts that *Emma* does not teach selection logic for causing a portion of instruction bytes of an instruction cache line selected by a fetch address not to be provided to an instruction buffer based on offset information specifying a location of a branch instruction within the cache line.

First, *Emma* teaches that his predicted branch instruction target address is used to select the next instruction to be loaded from the instruction buffer into a next instruction register (col. 15, lines 41-45), not to select a portion of the cache line bytes not to be stored to the instruction buffer as recited in claim 1. In other words, the implication is that *Emma* stores the entire cache line of instruction bytes (referred to by *Emma* as an instruction fetch segment) into the instruction buffer and then selects which instructions to decode on the way out of the instruction buffer, not on the way into the instruction buffer as recited by claim 1. See, e.g., col. 7, lines 5-19; col. 8, lines 23-29. This distinction is significant. Applicant's instant specification specifically states advantages of discarding instruction bytes before storing them into an instruction buffer. For example, one advantage is that the instructions may be more stored more efficiently in the instruction buffer. For another example, it may provide more time in the processor clock cycle for formatting or decoding instructions in the event of a critical timing path due to the alleviation of the need to include selection logic when taking instructions out of the instruction buffer.

Second, the Examiner cites the select logic, select gates, and construct block of Fig. 11 of *Emma* (collectively "selection logic" hereafter) as selection logic that causes a portion of the instruction bytes not to be provided to the instruction buffer. Fig. 11, elements 105, 106, and the "construct" block; col. 15, lines 4-45. Applicant respectfully asserts that *Emma*'s selection logic does not select instruction bytes from a cache line; rather, *Emma*'s selection logic selects branch prediction information. Specifically, *Emma*'s selection logic selects one of four sets of branch prediction information stored in an entry of *Emma*'s BHT. Element 82 of Fig. 11 illustrates the contents of an entry in *Emma*'s BHT. Col. 12, lines 53-58. *Emma* teaches that because a cache line is relatively large and may include multiple instructions, it is possible that multiple branch instructions may be present in the instruction cache line. Therefore, each entry in *Emma*'s BHT stores prediction information for up to four different branch instructions within a cache line.

Emma's selection logic of Fig. 11 is used to select the appropriate one of the four branch instructions to branch to and particularly to select the branch prediction information associated with the appropriate selected branch instruction. See col. 7, line 66 to col. 8, line 4; col. 15, lines 23-32; col. 12, line 62 to col. 13, line 3. Therefore, *Emma*'s "selection logic" does not select instruction bytes.

For the reasons stated above, Applicant respectfully asserts that *Emma* does not anticipate claim 1.

With respect to claim 2, the Examiner asserts that *Emma* teaches offset information provided by a branch target address cache that specifies a location of an instruction immediately following the branch instruction within the line of instruction bytes. Applicant respectfully asserts that *Emma* does not teach offset information that specifies a location of an instruction immediately following the branch instruction within the line of instruction bytes. *Emma* specifically states the information stored in his BHT entry identifies the address of each taken <u>branch</u> contained in the associated entry, not the address of the instruction immediately following the branch. Col. 12, lines 53-56.

With respect to claim 3, Applicant respectfully asserts that in the case of a "ghost hit", *Emma* teaches that the <u>entire</u> cache line specified by the predicted target address is not stored into the instruction buffer, rather than only a portion of the instruction bytes not being stored into the instruction buffer as recited by claim 3. Fig. 12; col. 15, lines 65 to col. 16, line 2; col. 16, lines 37-40: if BHT-HIT, but T=0, i.e., ghost-hit, then although the predicted target address (TA) is stored in the BA/TA stack, the instruction fetch address is not updated with the predicted TA. Thus, *Emma* does not teach the recited limitations of claim 3. Nevertheless, the entire cache line specified by the predicted target address of *Emma* is not the cache line containing the branch instruction as recited by claim 3, but rather is the cache line containing the target instructions of the branch instruction, i.e., the instructions specified by the predicted target address.

With respect to claim 4, Applicant respectfully asserts that as discussed above with respect to claim 1, element 82 of Fig. 11 of *Emma* is an entry in *Emma*'s BHT that stores branch prediction information, not instruction bytes.

With respect to claim 5, Applicant respectfully asserts that *Emma*'s valid bits denote whether the respective prediction information stored in the entry of his BHT is valid, not whether instruction bytes are valid. Again, this is because *Emma*'s BHT does not store instruction bytes, but store branch prediction information.

With respect to claim 18, Applicant respectfully asserts that *Emma* does not teach an instruction buffer directly coupled to instruction format logic as recited by claim 18. Fig. 10 clearly illustrates a next instruction register coupled between the instruction buffer and the instruction decoder. Contrary to the Examiner's assertion, Applicant can find no teaching in *Emma* of the next instruction buffer preparing the next instruction for decoding. *Emma* simply teaches that the next instruction register contains the next instruction to be processed by the instruction decoder. Col. 14, lines 21-23.

With respect to claim 23, Applicant respectfully asserts that *Emma* does not teach the recited limitations for the reasons discussed below with respect to claim 26.

Applicant respectfully asserts *Emma* does not anticipate dependent claims 2-25 because they depend from independent claim 1, which is not anticipated by *Emma* for the reasons discussed above.

With respect to claim 26, the Examiner asserts that *Emma* teaches selection logic that writes the branch instruction and target instruction immediately adjacent to one another into the instruction buffer. Applicant respectfully asserts that *Emma* does not teach selection logic that writes the branch instruction and target instruction immediately adjacent to one another into the instruction buffer for the reasons discussed above with respect to claim 1. In particular, the BA/TA pair cited by the Examiner are not instruction bytes; rather, they are addresses, in particular, the address of the branch instruction being predicted and the target address predicted by the BHT, respectively. Furthermore, the select logic, select gates, and construct block of Fig. 11 do not select instruction bytes; rather, they selection branch prediction information, such as the BA/TA pair and taken/not taken prediction (T) of the BHT entry 82.

Applicant respectfully asserts *Emma* does not anticipate dependent claims 27-37 for reasons similar to those discussed above with respect to claims 2-25, and because they depend from independent claim 26, which is not anticipated by *Emma* for the reasons discussed above.

With respect to claim 38, the Examiner asserts that *Emma* teaches providing to an instruction buffer portions of first and second cache lines remaining after discarding from the first cache line instructions after the branch instruction and discarding from the second cache line preceding the target instruction. Applicant respectfully asserts that *Emma* does not teach providing to an instruction buffer portions of first and second cache lines remaining after discarding from the first cache line instructions after the branch instruction and discarding from the second cache line preceding the target instruction for the reasons discussed above with respect to claim 1. In particular, the Examiner asserts it is inherent that any other instructions sequentially following the branch instruction will be discarded and that any instructions before the target instruction must be discarded because the target instruction is the next instruction to be executed. While it is true that instructions after the branch instruction and instructions before the target instruction will be discarded from execution, it does not necessarily follow that the instructions will be discarded prior to being stored into the instruction buffer. In other words, it may be inherent that *Emma* discards the instructions before they reach the execution stages of *Emma* pipeline, but not before reaching the instruction buffer. As discussed above with respect to claim 1, *Emma* teaches either storing or not storing an entire cache line, i.e., all the instructions of the cache line, into the instruction buffer, and subsequently selecting only certain instructions to remove from the instruction buffer at decode time based on the branch prediction information. Hence, *Emma* does not teach not storing remaining non-discarded portions of cache lines into the instruction buffer as recited by claim 38.

Applicant respectfully asserts *Emma* does not anticipate dependent claims 39-46 for reasons similar to those discussed above with respect to claims 2-25, and because they depend from independent claim 38, which is not anticipated by *Emma* for the reasons discussed above.

Rejection Under 35 USC 103

Claims 16-17 and 33-37

The Examiner rejected claims 16-17 and 33-37 under 35 U.S.C. § 103, as being unpatentable over *Emma* in view of *Schwendinger*, U.S. Patent No. 6,250,821 (hereinafter *Schwendinger*). Applicant respectfully traverses the rejection of claims 16-17 and 33-37.

With respect to claims 17 and 34, the Examiner asserts that *Schwendinger* teaches an instruction buffer that is at least a byte wide. Applicant has amended claims 17 and 34 to clarify that the instruction is exactly one byte wide. In contrast, *Schwendinger* teaches shifting out entire 32-bit wide instructions at a time, not single bytes. Therefore, *Emma* in view of *Schwendinger* does not obviate claims 17 and 34.

Applicant further respectfully asserts *Emma* in view of *Schwendinger* does not anticipate or obviate dependent claims 16-17 and 33-37 because they depend from independent claim 1 and 26, respectively, which are not anticipated or obviated by *Emma* for the reasons discussed above.

Claims 20 and 25

The Examiner rejected claims 20 and 25 under 35 U.S.C. § 103, as being unpatentable over *Emma* in view of *Miller*, U.S. Patent No. 6,081,884 (hereinafter *Miller*). Applicant respectfully traverses the rejection of claims 20 and 25. Applicant respectfully asserts *Emma* in view of *Miller* does not anticipate or obviate dependent claims 20 and 25 because they depend from independent claim 1, which is not anticipated or obviated by *Emma* for the reasons discussed above.

The Examiner has indicated additional prior art which is made of record and not relied upon. None of these references anticipate or obviate applicant's invention.

Applicant respectfully requests that a timely Notice of Allowance be issued in this case.

Applicant earnestly requests the examiner to telephone him at the direct dial number printed below if the examiner has any questions or suggestions concerning the application.

Respectfully submitted,

James W. Huffman
Huffman Law Group, P.C.
Registration No. 35,549
Customer No. 23669
1832 N. Cascade Ave.
Colorado Springs, CO 80907
719.475.7103
719.623.0141 fax
jim@huffmanlaw.net

Date: 9-20-04

"EXPRESS MAIL" mailing label number ED041359504US. Date of Deposit 9·20-04. I hereby certify that this paper is being deposited with the U.S. Postal Service Express Mail Post Office to Addressee Service under 37 C.F.R. §1.10 on the date shown above and is addressed to the U.S. Commissioner of Patents and Trademarks, Alexandria, VA, 22313.

By: Vicki J Hogan